

UNIVERZA V LJUBLJANI

Fakulteta za elektrotehniko

Stanislav Kovačič

Komunikacije v avtomatiki

(študijsko gradivo)

Ljubljana, 25. oktober 2001

6 Industrijska omrežja

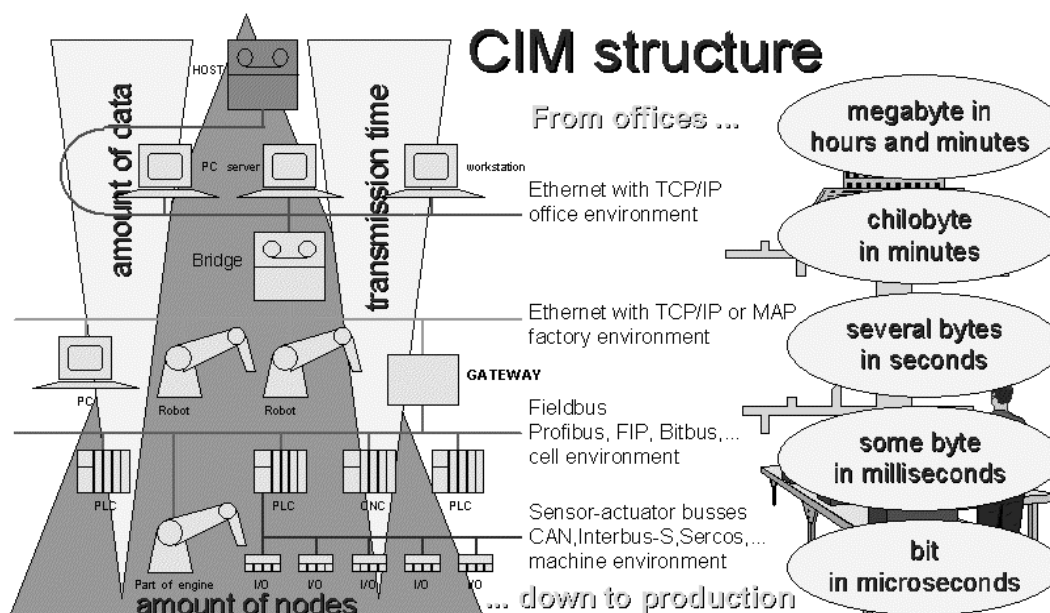
Industrijska omrežja so prilagojena potrebam, ki jih narekuje industrijsko okolje. Največkrat imajo obliko vodila, ali pa jim je vodilo kot način povezovanja vsaj za osnovo. Taka omrežja imajo precej dobrih lastosti, na primer omogočajo splošno ali skupinsko naslavljanje, in nudijo ekonomičen način povezovanja velikega števila razmeroma enostavnih naprav. Skupno ime za omrežja takega tipa je v angleškem jeziku Fieldbus, v nemškem Feldbus, mi pa jih bomo imenovali področna vodila. (Procesna) področna vodila so začela pridobivati na pomenu z razvojem mikroelektronske tehnologije, ki je omogočila gradnjo vse bolj sposobnih ("inteligentnih") in zanesljivih procesnih naprav. Inteligentni senzorji, aktuatorji, krmilniki in podobne naprave lahko prevzamejo nase številne funkcije v zvezi z obdelavo signalov, ki bi se sicer morale opravljati v centralnem računalniku. Sposobni pa so tudi preverjanja pravilnosti lastnega delovanja, parametriranja (umerjanja) na daljavo, in podobno. Analogne veličine se ne prenašajo analogno (npr. z 4-20 mA merilno zanko), kot na primer v telemetrijskih sistemih, temveč se takoj digitalizirajo, po potrebi predobdelajo in digitalno tudi prenašajo. Porazdeljeno procesiranje skupaj z digitalnim načinom prenosa zagotavlja večjo zanesljivost in prilagodljivost ob manjših stroških.

Procesna področna vodila so v marsičem zelo podobna lokalnim omrežjem, od njih pa se značilno razlikujejo v pomenu informacije, ki jo prenašajo, in v okoljih, kjer se uporabljajo. V tej podobnosti in različnosti z omrežji tipa LAN gre najbrž iskati razlog, da je eden od proizvajalcev omrežij za procesne namene (Echelon) skoval kratico LON (ang. Local operating Networks). Med omrežja tipa LON bi lahko šteli omrežja LAN, ki so bila prirejena za delovanje v stvarnem času in omrežja, ki so bila razvita nalašč za potrebe industrijske informatizacije. Če je v omrežjih LAN pomembnejša prepustnost od odzivnosti, pa se v omrežjih LON poudarja kratek odzivni čas in zmožnost prioritetnega obravnavanja prenašanih podatkov.

Mrežnih arhitektur in tehnologij, ki jih lahko uvrstimo med procesna področna vodila, je veliko. Tipični predstavniki področnih vodil so Profibus, WorldFIP, P-Net, Interbus-S, CAN, Bitbus, ControlNet, DeviceNet, SDS, Modbus Plus, i.t.d., od katerih ima vsak še podizvedbe, prilagojene specifičnim potrebam določene industrije. Čeprav je pestrost ponudbe in izbire zaželena, pa je po drugi strani področnih vodil tudi preveč, še posebno ko se pokaže potreba po povezovanju naprav in podsistemov različnih proizvajalcev. V praksi se izkaže, da tako imenovana odprtost sistemov največkrat sama po sebi še ni dovolj.

Slika 117 prikazuje, kako se komunikacijska tehnologija vklaplja v model računalniško integrirane proizvodnje (CIM - Computer Integrated Manufacturing). Na sezorskem nivoju imamo razmeroma malo podatkov (signalov), vednar zahtevajo kratek odzivni čas. Proti vrhu piramide količina podatkov narašča,

vendar ti podatki niso časovno kritični (dopuščajo daljše odzivne čase). Visoka prepostnost omrežja je zaželena, a ni obvezna.



Slika 117: Komunikacijska tehnologija v računalniško integrirani proizvodnji.

Vsa omrežja, ki se uporabljajo za prenos procesnih podatkov (precesnih v širšem smislu besede), niso po obliki vodilo. Vendarle pa omrežja te oblike prevladujejo ali pa imajo z vodilom nekaj skupnega. V tabeli 118 vidimo načelno razvrstitev vidnejših predstavnikov mrežnih tehnologij na spodnjih nivojih industrijske informatizacije. Spodnji nivo je podrobneje razslojen na senzorski nivo, nivo naprav in na področni nivo. Vidimo, da nekatere tehnologije, kot na primer LonWorks segajo preko več podnivojev.

Sensor-Actuator Bus (Bit-level)	Device Bus (Byte-level)	Fieldbus (Block-level)
CAN	CAN	IEC 1158/ISA SP50.02
AS-Interface	DeviceNet	FOUNDATION Fieldbus
InterBus Sensor Loop	SDS	Profibus-PA
Ser iplex	CAL/CANopen	Profibus-FM
SERCOS	CAN Kingdom	WorldFIP
Sensoplex	InterBus-S	P-NET
	Device WorldFIP (DWF)	Measurement Bus
	FIP IO	Bitbus
	Profibus-DP	
	I/O Lightbus	
	SERCOS	
	MIL-STD-1553	
	LonWorks	

Slika 118: Razvrstitev mrežnih tehnologij po namenu uporabe.

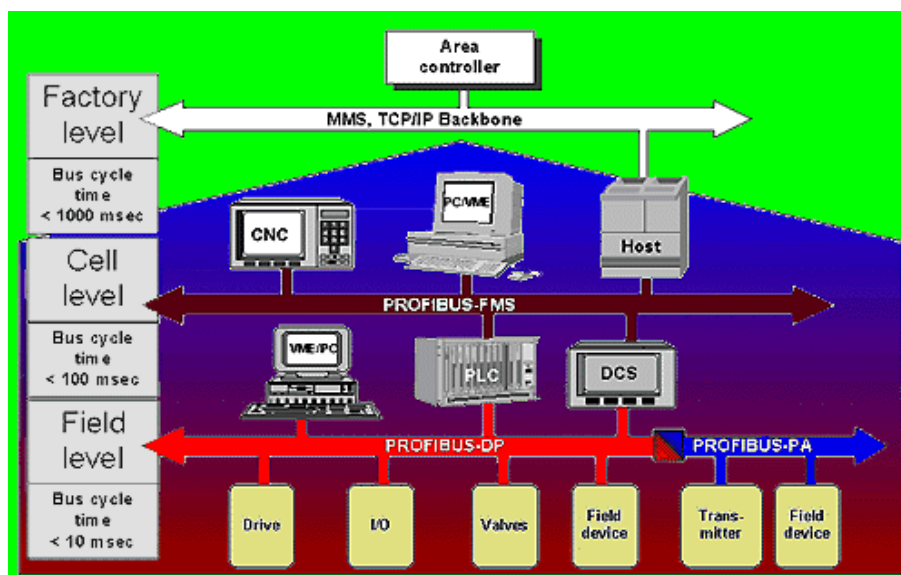
Naslednji seznam zaobsega nekaj najbolj razširjenih omrežnih tehnologij in izvedbe področnih in podobnih omrežij:

- IEC/ISA SP50, mednarodni Fieldbus standard, ki obsega štiri sloje: fizični, podatkovni, aplikacijski, uporabniški (PL, DLL, AL, UL) in upravljanje omrežja.
- Foundation Fieldbus (FF) je področno vodilo fundacije: Fieldbus Foundation.
- WorldFIP: World Factory Instrumentation Protocol, ki je nastal na podlagi francoskega nacionalnega standarda FIP.
- Profibus: PROcess FIEldBUS (verzije FMS, DP, PA) so razvili v Nemčiji in razširjen po vsem svetu, posebno po zaslugi firme Siemens. Od leta 1996 evropski standard EN 50 170.
- P-NET: Process automation NET je omrežje danskega porekla (PROCES-DATA Silkeborg ApS). Od leta 1996 evropski standard EN 50 170.
- INTERBUS-S je razvila firma Phoenix Contact za povezovanje na nivoju senzorjev in aktuatorjev.
- CAN: Controller Area Network so razvili v Nemčiji v glavnem pri firmi Rober Bosch GmbH, sprva za potrebe avtomobilske industrije. Je tudi mednarodni standard ISO 11898 in ISO 11519-2.
- DeviceNet je razvila firma Allen-Bradley; temelji na CAN. Podpira ga ODVA (Open DeviceNet Vendor Association), Inc.
- SDS: Smart Distributed System so razvili pri Honeywell's MICRO SWITCH Division, temelji na CAN.
- Bitbus je razvil Intel leta 1984. Kasneje je bil dopolnjen in standardiziran z dokumentom IEEE 1118.
- LonWorks: Local Operating Network so razvili pri Echelon-u. Vsebuje vse sloje ISO/OSI modela.
- Modbus, Modbus Plus sta omrežji, ki jih je razvila firma MODICON.
- ASI: Actuator Sensor Interface, razvit v Nemčiji s strani konzorcija dobaviteljev procesne opreme, za njim stoji v glavnem Siemens.
- M-Bus: Meter Bus, sistem hišne elektronike (HES) za daljinsko branje gospodinjskih števec.

- CEBus: Consumer Electronic Bus, razvit s strani konzorcija proizvajalcev s pomočjo EIA (Electronic Industry Association).
- SERCOS (SErial Real-time COmmunication System) mednarodni standard IEC 1491 za komunikacijo med digitalnimi pogoni in kontrolnimi enotami numerično krmiljenih strojev, razvit v Nemčiji VDW (German Machine Tool Builders Association) in ZVEI (German Electrical Standards Association),
- ARCNET: Attached Resource Computer Network, razvit pri Datapoint in podprt s strani ARCNET Trade Association.
- HART (Highway Addressable Remote Transducer), razvit pri Rosemount in podprt od HCF (HART Communication Foundation). Ni pravo področno vodilo, v glavnem DDL (Device Description Language).

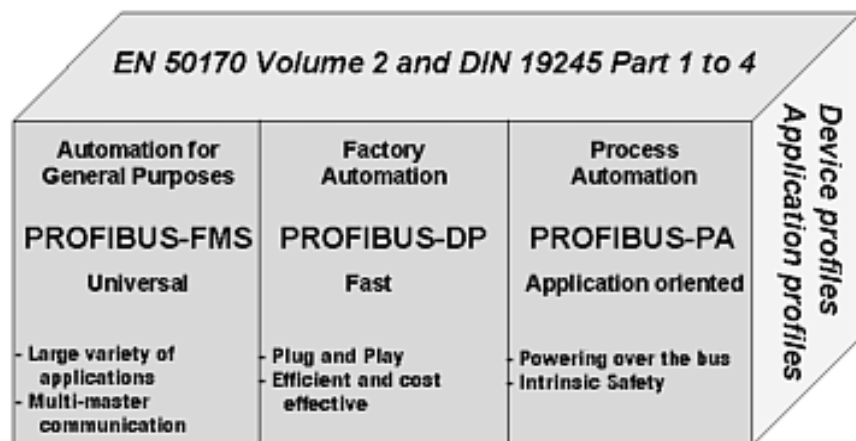
6.1 Profibus

Procesno področno vodilo Profibus (Process Field Bus) je nastalo konec 80 let v Nemčiji. Profibus med področnimi vodili za enkrat vodi z največ inštalacijami na svetu. Glavni zagovornik omrežja Profibus je Siemens. Profibus je zasnovan na podlagi obstoječih nemških in mednarodnih standardov. Upošteva arhitekturni model ISO OSI, vendar definira samo prvi, drugi in sedmi sloj. Lahko bi dodali, da nad sedmim slojem nadgradi od namena uporabe odvisen sloj.



Slika 119: Profibus v sistemih integrirane proizvodnje (Povzeto po: M. Volz Profibus Technical Overview)

Slika 119 prikazuje družino omrežij Profibus in kako se omrežja vklaplajo v hierarhijo računalniško integrirane proizvodnje, slika 120 pa podaja njihove glavne značilnosti z namenom uporabe in oznako ustreznega standarda. Obstajajo tri različice omrežja Profibus, in sicer: Profibus-FMS, Profibus-DP in Profibus-PA.



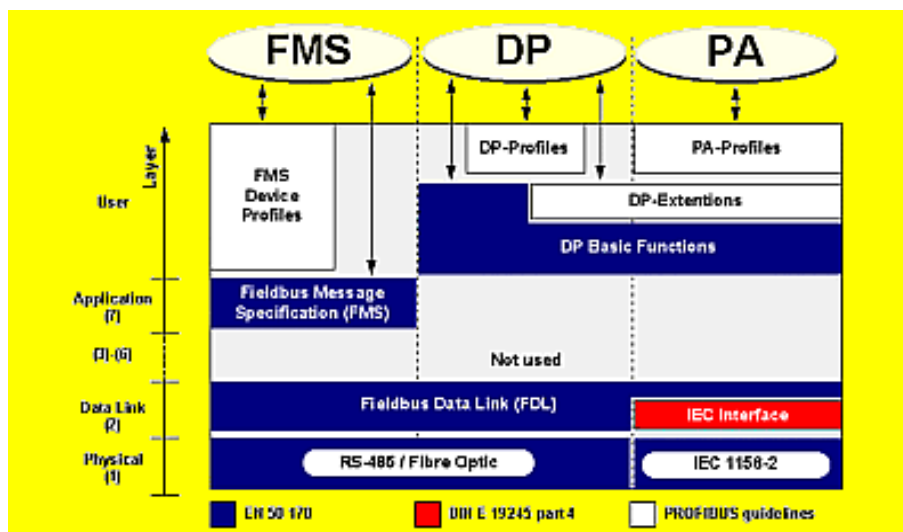
Slika 120: Družina Profibus, njene glavne značilnosti z namenom uporabe in ustreznim standardom. (Povzeto po: M. Volz Profibus Technical Overview)

Profibus-FMS je splošno namensko omrežje na celičnem nivoju. Odlikuje ga velika prilagodljivost za potrebe konkretne aplikacije. Definira fizični in podatkovni sloj ter sloj aplikacije. Sloji 3,4,5 in 6 niso izraženi. Funkcije teh slojev so zajete v podsloju LLI (ang. Lower Layer Interface), ki je del sloja aplikacije. FMS (ang. Fieldbus Message Specification) definira protokol aplikacije in realizira storitve, ki jih ta sloj opravlja za uporabnika. FMS definira tudi vmesnik z uporabnikom. Storitve FMS so definirane kot podmnožica funkcij MMS (Manufacturing Message Specification - ISO 9506) omrežja MAP (Manufacturing Automation Protocol), optimiranih za potrebe področnih vodil.

PROFIBUS-PA je zasnovan posebej za potrebe procesne avtomatizacije v nevarnem okolju (kemična, petro kemična industrija). Omogoča prenos podatkov in napajanja po skupnem dvožilnem kablu v skladu z mednarodnim standardom IEC 1158-2.

Omrežje Profibus-DP je optimirano za povezovanje decenarilizirane periferije (DP) s centralnim krmilnikom. Definira samo spodnja dva sloja, medtem ko sloji 3-7 niso izraženi. Za preslikavo funkcij uporabniškega vmesnika na drugi sloj poskrbi DDLM (Direct Data Link Mapper). DDLM je specificiran v dokumentu DIN 19245 Teil3.

Kako se arhitektura Profibus vklaplja v ISO OSI arhitekturni model in njeno nadgradnjo na nivoju uporabne prikazuje slika 121.



Slika 121: Arhitektura Profibus in model OSI. (Povzeto po: M. Volz Profibus Technical Overview)

6.1.1 Fizični sloj

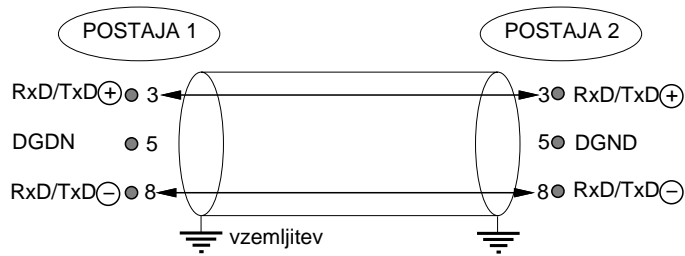
Podlaga za osnovno izvedbo fizičnega sloja je ameriški standard RS485, obstajajo pa tudi druge možnosti, kot IEC 1158-2 in optična vlakna. Za prenosno sredstvo služi oklopljena parica s karakteristično upornostjo 100 do 130 ohmov. Najvišja dovoljena hitrost prenosa pada z dolžino segmenta, vendar je dopustna dolžina odvisna tudi od kvalitete sredstva (kabela). Za dolžino segmenta 1200 metrov sme biti hitrost samo še 93.75 Kb/s. Najvišja dovoljena hitrost je 12 Mb/s in sicer pri razdalji 100 metrov.

Hitrost [Kb/s]	9.6	19.2	93.75	187.5	500	1500	12000
Razdalja [m]	1200	1200	1200	1000	400	200	100

Na en segment je lahko priključenih največ 32 postaj. Omrežje se lahko razširi s pomočjo ponavljalnikov do 127 postaj. Med dvema poljubnima postajama ne smejo biti več kot 3 ponavljalniki. Za priključitev postaj na vodilo se uporablja 9-polni konektor DB9, glej sliko 122.

6.1.2 Dostop do prenosnega sredstva

Dostop do vodila (MAC) združuje pozivanje s podajanjem žetona. Obstajata dva tipa postaj: aktivne in pasivne. Aktivne postaje ali gospodarji si v dogovorjenem zaporedju podajajo žeton. Na ta način tvorijo logični obroč, kot prikazuje slika 123. Postaje, ki niso v logičnem obroču, so podrejene oziroma pasivne postaje. Ko aktivna postaja dobi žeton, postane gospodar vodila, dokler ji ne poteče



Slika 122: Vezava postaj na omrežje PROFIBUS.

dodeljeni čas. Nato preda žeton naslednji aktivni postaji. Postaja, ki ima žeton, lahko s pozivanjem dobi ali pošlje podatke kateri koli pasivni ali aktivni postaji. Pasivna postaja sama od sebe ne sme začeti z oddajo.

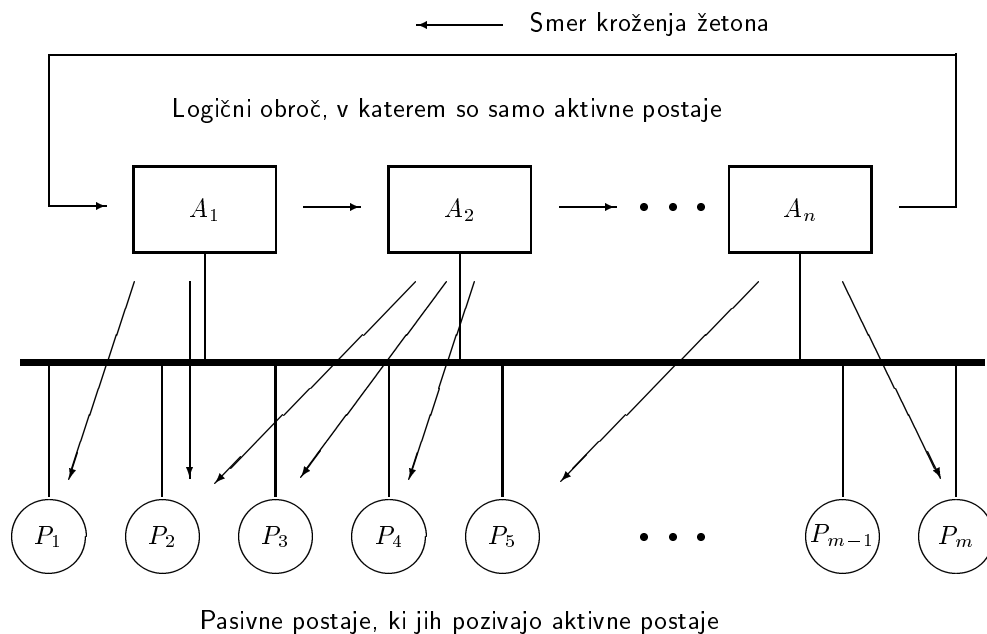
Slika 123 prikazuje splošno konfiguracijo omrežja. Možno je tudi čisto vodilo z žetonom brez pasivnih postaj ali pa čisto pozivanje z eno samo aktivno in več pasivnimi postajami, ki jih poziva aktivna postaja. Običajno je v omrežju relativno malo aktivnih postaj. Na ta način se skrajša obhodni čas žetona. Kratek obhodni čas žetona, skupaj s prioritetenim obravnavanjem pomembnejših podatkov, pa zagotavlja kratek odzivni čas.

Profibus-DP v načelu predvideva samo eno aktivno postajo, ki krožno poziva več pasivnih postaj (Monomaster sistem – tip aktivne postaje DPM1). V enem pozivnem ciklu vsem postajam pošlje in od njih sprejme sveže podatke. Če je v omrežju več aktivnih postaj (tip postaje DPM2), te opravljajo upravljalne funkcije. V primeru, da je v omrežju več aktivnih postaj tipa DPM1, pa smejo pasivne postaje sprejemati podatke samo od ene aktivne postaje (definirano s konfiguracijo), bere pa jih lahko tudi več aktivnih postaj.

6.1.3 Prioritete

Profibus nudi na podatkovnem sloju dva prioriteta nivoja okvirjev ali telegramov: višjo in nižjo. Kadar dobi aktivna postaja žeton, ima vedno možnost oddati vsaj en okvir višje prioritete, okvirje z nižjo prioriteto pa sme oddajati samo v primeru, če ji po oddaji prioritetenih okvirjev prej ne poteče razpoložljivi čas imenovan držalni čas žetona (T_{TH}). Prioriteto sporočil določa uporabnik. Zato, da se doseže zeleni učinek (hiter odziv za pomembne podatke), uporabnik ne sme pretiravati s številom prioritetenih sporočil.

V procesu kroženja žetona sta pomembna naslednja časovna parametra: ciljni obhodni čas žetona T_{TR} (Target Token Rotation Time) in dejanski obhodni



Slika 123: Topologija omrežja Profibus in način dostopa do prenosnega sredstva (vodila).

čas žetona T_{RR} (Real Token Rotation Time). T_{RR} je parameter, ki je odvisen od zahtev in konfiguracije omrežja. Ko postaja dobi žeton, postavi časovnik T_{RR} na nič, ki nato prosto teče. Ko postaja ponovno dobi žeton, sme vedno, brez omejitev, oddati eno sporočilo višje prioritete. Potem primerja dejanski obhodni čas T_{RR} z dovoljenim T_{TR} . V primeru, da je držalni čas žetona (Token Holding Time) $T_{TH} = T_{TR} - T_{RR} > 0$, sme postaja čas T_{TH} oddajati druge okvirje, potem pa mora brezpogojno predati žeton. Okvir, ki je v procesu oddajanja, bo oddan do konca, vključno z morebitnimi ponovitvami oddaje tudi v primeru, da T_{RR} preseže T_{TR} , vendar se presežek direktno upošteva za skrajšanje držalnega časa ob naslednjem prejemu žetona.

6.1.4 Minimalni ciljni obhodni čas žetona

Najkrajši ciljni obhodni čas žetona T_{TRmin} je odvisen od števila aktivnih n_a udeležencev, od trajanja žetona T_{TC} , od trajanja okvirja z visoko prioriteto T_{MC-h} , od trajanja okvirja z nizko prioriteto T_{MC-l} in števila teh okvirjev na en obhod žetona n_p ter od (pričakovanega) časa potrebnega za ponovitev sporočil na en obhod žetona T_{MC-r} . T_{TRmin} ocenimo po sledeči formuli:

$$T_{TRmin} = n_a \times (T_{TC} + T_{MC-h}) + n_p \times T_{MC-l} + T_{MC-r}$$

Pri tem smo predpostavili enako dolžino prioritetnih okvirjev, enako dolžino okvirjev nižje prioritete in da ima vsaka aktivna postaja možnost oddati en visoko

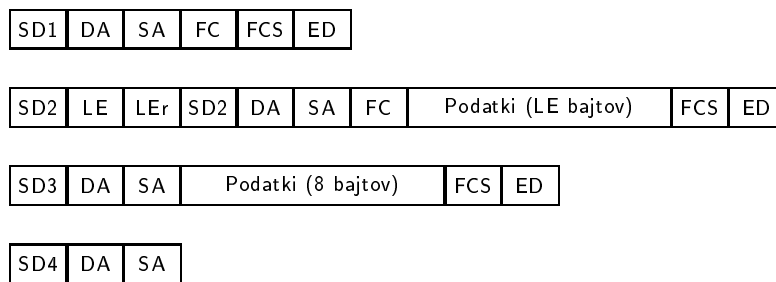
prioritetni okvir na obhod žetona. V slučaju, da ni ponovitev, določa prvi člen maksimalni odzivni čas za okvirje z visoko prioriteto. Drugi člen daje možnost tudi okvirjem z nižjo prioriteto. V splošnem si želimo, da bi bil čas T_{TR} čim krajši. Zato naj bi bilo aktivnih postaj kot tudi prioriternih okvirjev malo, prioritetni okvirji pa kratki (npr. največ 20 bajtov podatkov).

6.1.5 Upravljanje obroča

Žeton kroži med aktivnimi udeleženci od postaje z nižjim proti postajam z višjimi naslovi. Postaja z najvišjim naslovom (ta naslov je znan vsem postajam) pošlje žeton postaji z najnižjim naslovom in obroč je sklenjen. Vsaka postaja ima svojega predhodnika z nižjim naslovom, od katerega pričakuje žeton in naslednika z višjim naslovom, ki mu preda žeton. Aktivni udeleženci so zabeleženi v seznamu aktivnih postaj (LAS - List of Active Stations), s katerim razpolaga vsaka aktivna postaja. V kolikor neka postaja zazna na njo naslovljen žeton, ki ga ne pošilja njen predhodnik, predpostavlja, da gre za napako in žetona ne upošteva. Šele po ponovnem sprejemu žetona od iste postaje razume to kot spremembo obroča, zato prevzame žeton in obnovi LAS.

Ko postaja preda žeton nasledniku, istočasno opazuje aktivnost na vodilu (s tem preveri delovanje tudi lastnega oddajnika). V kolikor v predvidenem časovnem intervalu pride do nepričakovane aktivnosti na vodilu (slediti mora okvir z izvornim naslovom naslednika) interpretira to kot napako. Zato se umakne iz obroča in gre v začetni način delovanja, ki zahteva ponovno vabilo v obroč ali celo ponovno vzpostavitev obroča. V kolikor v predvidenem času ne pride do pričakovane aktivnosti (vodilo je prosto), poskusi ponovno s predajo žetona. Če tudi tokrat predaja ne uspe, poskusi s predajo nasledniku njegovega naslednika. Če niti ta poskus ne uspe, postaja predpostavi, da je obroč razpadel in prevzame nalogo ponovne vzpostavitve obroča. V ta namen pošlje žeton dvakrat zapored na samo sebe in s tem obvesti ostale udeležence, da začnjo ponovno vzpostavitev obroča.

Aktivne postaje smejo svobodno vstopiti v obroč ali iz njega izstopiti. Aktivni udeleženec, ki je v obroču, je pooblaščen za povabilo za vstop v obroč tistih postaj, ki so v naslovnem območju med njim in njegovim naslednikom. To naslovno področje se imenuje naslovna reža. Osvežitev seznama naslovne reže (npr. vabilo za vstop) se opravlja ciklično, za kar skrbi časovnik časa T_{GAP} (GAP Update Time). Postaja, ki ima žeton in se ji po oddaji sporočil še ni iztekel držalni čas žetona, poizveduje zapored po stanju postaj znotraj adresne reže. V kolikor se pasivna postaja ne javi niti na drugi poziv, se izvzame iz seznama. V kolikor aktivna postaja odgovori, da je pripravljena na vstop v obroč, mu postaja preda žeton in s tem je njeno delo opravljeno.



Slika 124: Oblike okvirjev podatkovnega sloja Profibus.

Vzpostavitev ali ponovna vzpostavitev obroča je potrebna tedaj, kadar v obroču ni žetona (torej obroč niti ne obstaja). Vzpostavitev obroča je pravzaprav poseben primer osvežitve seznamov aktivnih udeležencev in seznama adresne reže. Vzpostavitev obroča začne postaja, ki odda žeton dvakrat zapored na samo sebe. Nato začne s povpraševanjem postaj od nižjih proti višjim naslovom. V kolikor se postaja predstavi kot pasivna, se vpiše v GAPL. V seznam GAPL se vpiše tudi aktivna postaja, ki še ni pripravljena na vstop v obroč. Postaja nadaljuje s poizvedovanjem dokler ne naleti na prvo aktivno postajo, ki je pripravljena na vstop v obroč in ji preda žeton.

6.1.6 Oblike okvirjev (“telegramov”)

Profibus pozna na podatkovnem sloju naslednje oblike okvirjev:

- okvirji s konstantno dolžino brez podatkov: zahteva, potrdilo, in kratko potrdilo,
- okvirji s konstantno dolžino s podatki: zahteva s podatki ali odziv s podatki, in
- okvirji s spremenljivo dolžino: zahteva s podatki in potrdilo s podatki.

Naslednje slike prikazujejo oblike naštetih okvirjev.

Med zaporednima okvirjema je potreben premov v trajanju najmanj 33 bitov. Okvir začne z začetnim znakom SD1, SD2, SD3 ali SD4 (Starting Delimiter), ki napove začetek okvirja in hkrati kodira njegovo obliko. SD1 (\$10) je oznaka okvirja brez podatkov, SD3 (\$A2) je oznaka okvirja s podatki, SD2 (\$68) je oznaka okvirja spremenljive dolžine. Žeton ima oznako SD4 (\$DC). Znak ED (Ending Delimiter) (koda \$16) označuje konec okvirja. Polje FC (Frame Control)

kodira pomen okvirja (na primer pozitivno ali negativno potrdilo, tip postaje: aktivna ali pasivna, tip storitve, i.t.d.). DA (Source Address) in DA (Destination Address) sta naslova izvirne in ponorne postaje. FCS (Frame Check Sequence) je za preverjanje pravilnosti prenosa, ki zagotavlja Hammingovo razdaljo $HD = 4$. V okvirjih spremljive dolžine je LE (Length) dolžina – število podatkov v okvirju (do 246 bajtov) in LER je ponovljena dolžina zaradi zanesljivosti. Potrditev okvirja je možna tudi z enim samim znakom SC, ki ima vrednost \$ E5. To daje možnost hitri potrditvi, kadar ni potrebe za prenos podatkov v nasprotni smeri.

6.1.7 Storitve linijskega sloja

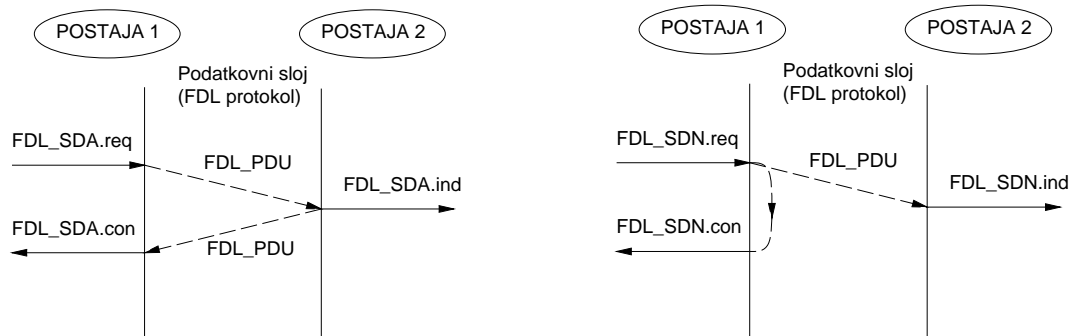
Podatkovni/linijski sloj nudi na podlagi protokola FDL (ang. Fieldbus Data Link) sloju aplikacije naslednje storitve:

- SDA (Send Data with Acknowledge) - pošlji podatke in zahtevaj potrditev,
- SDN (Send Data with no Acknowledge) - pošlji podatke brez zahteve za potrditev,
- SRD (Send and Request Data with Reply) - pošlji podatke in zahtevaj odgovor s podatki,
- CSRD (Cyclic Send and Request Data with ciklično pošiljanje podatke in zahtevaj odgovor s podatki.

Časovni potek storitev SDA in SDN je skiciran na sliki 125. SDA je torej potrjena storitev in se uporablja za pošiljanje podatkov, pri čemer mora sprejemna stran sprejem podatkov obvezno tudi potrditi. Oddajna stran v vsakem primeru dobi potrdilo - pozitiven ali negativen odgovor. Za razliko od osnovnega modela potrebne storitve, SDA storitev nima odziva. Odziv pride od aktivnosti linijskega sloja oddaljene strani. Ob tej priliki ponovno povejmo, da je storitev eno, protokol pa drugo. Potrdilo storitve (ang. Confirmation) ne smemo zamenjevati s potrdilom (ang. Acknowledgment) kot sestavine protokola. Popolnoma jasno je, da je za prenos protokolovne podatkovne enote (PDU) v splošnem lahko potrebnih več poskusov, kar vključuje \pm potrditve.

Storitev SDN ni potrjena. Uporablja se največ pri skupinskem in splošnem naslavljanju, to je pošiljanju istih podatkov na več naslovov (ang. Broadcast in Multicast). Oddajna stran dobi samo lokalno potrditev, kar ji seveda ne zagotavlja, da je oddaljena stran podatke v resnici tudi dobila.

Storitev SRD je najsplošnejši način za pošiljanje in istočasno zbiranje podatkov. Možna sta oba podprimera, to je, ko se podatki samo pošiljajo ali samo



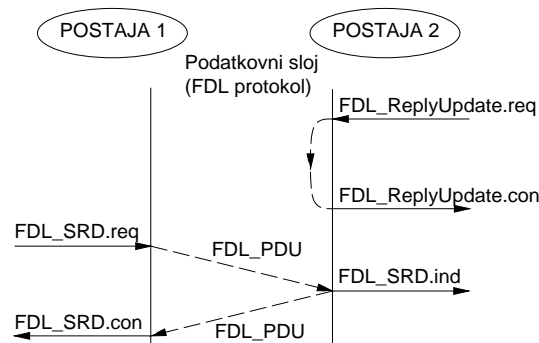
Slika 125: Storitvi SDA in SDN podatkovnega sloja omrežja Profibus. SDA (levo) je potrjena storitev; potrditev generira podatkovni sloj na oddaljeni strani (Postaja 2). SDN (desno) je nepotrjena storitev; potrditev, ki jo dobi jemalec storitve, generira podatkovni sloj na lokalni strani (Postaja 1). Potrditev je zgolj lokalna.

zbirajo. V tem primeru je podatkovno polje okvirja (podatkovna enota storitve) v eni ali drugi strani prazno. Potek storitve je skiciran na sliki 126.

Storitev CSRSD se uporablja za ciklično pošiljanje podatkov na skupino naslovov in po potrebi istočasno zbiranje podatkov. S to storitvijo je na nivoju podatkovnega sloja realizirano krožno pozivanje postaj (Polling), z namenom, da se podatki tem postajam pošljejo ali od njih zberejo. Sloj aplikacije na lokalni strani zahteva z FDL_CSRSD.req od podatkovnega sloja ciklično storitev ter mu skupaj z zahtevo preda seznam naslovov postaj na katere želi poslati ali od njih dobiti podatke. Nato mu zapored predaja podatke za posamezno postajo in zahteva oddajo. Če lokalna stran kot odgovor oddaljene strani dobi podatke (kot pri storitvi SRD), jih takoj preda sloju aplikacije ter nadaljuje s pozivanjem (nadaljuje CSRSD storitev), dokler se zahteva eksplicitno ne razveljavi.

6.1.8 FMS – sloj aplikacije za Profibus-FMS

Protokole in storitve sloja aplikacije v omrežju Profibus-FMS pokrivata FMS (Fieldbus Message Specification) in LLI (Lower Layer Interface). LLI skrbi za prilagoditev sloja aplikacije na podatkovni sloj oziroma pravkar obravnavane storitve, ki jih ta nudi. FMS ima za specifikacijo MMS (Manufacturing Message Specification), ki je kot mednarodni standard opredeljen v dokumentih ISO/IEC 9506-1 in ISO/IEC 9506-2. ISO/IEC 9506-1 opisuje storitve aplikacijskega vmesnika in ISO/IEC 9506-2 definira protokol aplikacijskega sloja. FMS realizira podmnožico funkcij FMS, optimiranih za področje uporabe Profibus-FMS (predvsem povezovanje PLC). Bistvenega pomena je gotovo koncept odje-



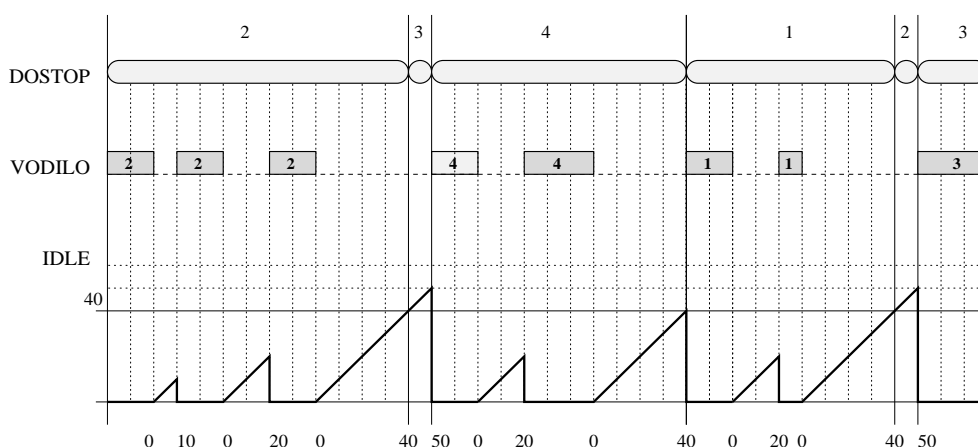
Slika 126: Storitve SRD podatkovnega sloja omrežja Profibus. Aktivnost podatkovnega sloja na oddaljeni strani (Postaja 2) pošlje v odgovor sveže podatke. Če podatkov nima, pošlje samo potrdilo. Sveže podatke dobi podatkovni sloj od sloja aplikacije preko storitve `FDL_ReplyUpdate`.

malec/strežnik (ang. Client/Server Model) skupaj s konceptom navidezne (virtualne) področne naprave VFD (ang. Virtual Field Device) oziroma v sklopu MMS proizvodne naprave VMD (Virtual Manufacturing Device). Strežnik je naprava, ki daje svoja sredstva (dejavnosti, podatke) na voljo drugim, tipično oddaljenim procesom, to je odjemalcem. Odjemalec (npr. uporabnikov proces) postavi zahtevo za izvedbo potrebne storitve, ta zahteva se po omrežju prenese do strežnika (strežnega procesa). Strežnik poskrbi za izvedbo storitve in se preko omrežja odzove odjemalcu, ki je potem obveščen o realizaciji (uspešni ali neuspešni) storitve. Po konceptu FMS navidezno napravo realizira strežnik. Navidezna naprava daje izgled resnične naprave z vidika ča krajevno porazdeljene aplikacije. Daje izgled resnične naprave (npr. ventila, merilnika tlaka) s stališča medprocesne komunikacije. Na ta način se doseže enotno obravnavanje različnih naprav, neodvisno od njihove izvedbe, če so s stališča zahtevane funkcije ekvivalentne.

6.2 P-Net

P-Net so razvili na Danskem v začetku osemdesitih let (1983), za povezovanje industrijskih naprav na področnem nivoju. Od marca 1996 je tudi evropski standard EN 50170 (Part 1). Dostop do vodila v tem omrežju ima nekaj zanimivih lastnosti, ki nas bodo zanimale. Še prej pa pogledjmo nekaj splošnih podatkov. Omrežje P-Net ima obliko vodila in temelji na standardu RS485. Na en segment gre do 125 postaj. Standardna hitrost prenosa je 76800 b/s. Podobno kot Profibus sta v omrežju P-Net dva tima postaj: nadrejene (ang. Master) in podrejene (ang. Slave). Na enem segmentu je lahko do 32 nadrejenih postaj. V nekem obdobju

ima lastništvo nad vodilom samo ena od nadrejenih postaj. Ta postaja lahko tedaj poziva druge postaje – upravlja vodilo. Rečemo, da ima žeton. Nadrejene postaje druga za drugo prevzemajo vodilo in na ta način tvorijo obroč. V tem pogledu je omrežje nepristransko. Logični obroč je zaznovan na podajanju **virtualnega žetnona**; virtualnega žetona zato, ker v resnici kot okvir ne obstaja. Kroženje virtualnega žetona je urejeno z ugotavljanjem dolžine časovnega intervala, ko je vodilo prosto. V ta namen ima vsaka nadrejena postaja dva števec: PROSTO in DOSTOP. Števec PROSTO se povečuje za ena s hitrostjo oddajanja/sprejemanja. Števec začne šteti vedno od nič navzgor, vsakič ko postane vodilo prosto. Če ostane vodilo prosto vsaj 40 bitnih celic, se poveča števec DOSTOP za ena. Števec DOSTOP se poveča za ena vsakič, ko števec PROSTO doseže vrednost 40, 50, 60 in tako naprej. Ta števec teče od nič do števila nadrejenih postaj na vodilu (najvišje adrese). Lahko bi ga imenovali kar adresni števec. Postaja lahko prevzame vodilo (začne z oddajo) tedaj, ko se števec DOSTOP izenači z njenim naslovom. To pomeni, da je postaja dobila žeton. Ni pa nujno, da začne z oddajo. V tem primeru ostane vodilo prosto in števec PROSTO teče naprej. Vsakič, ko se poveča za deset, se adresni števec poveča za ena in pravico za oddajo dobi naslednja postaja. Ko doseže vrednost najvišje prisotne adrese, začne šteti spet od začetka. Možne razmere za štiri nadrejene postaje prikazuje slika 127. Adrese postaj so 1, 2, 3, in 4. Najprej ima vodilo postaja 2, adresni števec postaj ima vrednost 2. Postaja 2 komunicira z drugimi postajami. Vsakič, ko postane vodilo prosto, steče števec PROSTO. Ko ostane vodilo prvič prosto za več kot 40 enot, se poveča števec DOSTOP za ena in postaja 2 izgubi lastništvo vodila. Vodila bi se lahko polastila postaja 3, vendar se v narisanih razmerah to ne zgodi. Postaja 3 se ne polasti žetona. V naslednjih 10 enotah se števec DOSTOP poveča na 4. Tokrat se postaja 4 polasti vodila in vodi komunikacijo, ki se zaključi s premorom. Adresni števec se spet poveča in ker so v omrežju samo štiri nadrejene postaje, preplavi na 1, vendar ostane postaja 1 neaktivna. Adresni števec se poveča na 2 in vodila se polasti postaja 2.



Slika 127: Načelo virtualnega žetona v omrežju P-Net.

6.3 CAN

CAN (Controller Area Network) so razvili v sedemdesetih letih v nemški firmi Bosch za potrebe avtomobilske industrije. Glavni namen razvoja omrežja CAN je bil zmanjšanje števila povezav v vozilih nove dobe. Taka vozila vsebujejo vseveč elektronskih komponent (senzorjev, aktuatorjev), ki povečujejo varnost in udobnost vožnje in jih je potrebno povezati med seboj. Tovrstna omrežja morajo zadostiti številnim ostrim zahtevam, morda najvažnejša pa je hiter odziv, ki ga zagotavlja CAN. Kratek in predvsem predvidljiv odzivni čas je bistvenega pomena v vseh sistemih, ki zahtevajo delovanje v stvarnem času (prepozno delovanje zavor je lahko prav tako usodno kot odpoved), zato se je CAN hitro uveljavil tudi na drugih področjih in bil tudi standardiziran (ISO 11898 in ISO 11519-1). Poleg determinističnega odziva je v omrežjih tipa CAN učinkovito rešen sistem prioritete; sporočilo z višjo prioriteto pride vedno prej na vrsto kot sporočila z nižjo prioriteto.

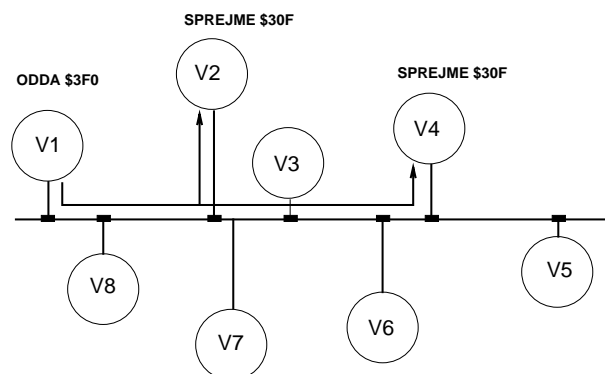
CAN kot tak se pretežno nanaša na drugi sloj oziroma spodnja dva sloja modela OSI. Definira način dostopa do skupnega prenosnega medija, medija samega pa ne definira. Drugi, višji sloji, so trenutno v razvojni fazi. Na primer, omrežja DeviceNet (Allen-Bradly) in SDS (Smart Distributed Sensors firme Honeywell) definirata sloj uporabe, a koristita CAN na sponjih dveh slojih.

Omrežje CAN je po obliki vodilo. Število postaj na vodilu teoretično ni navzgor omejeno, omejena pa je dolžina vodila. Z dolžino vodilo narašča čas širjenja signala od konca do konca, ta pa je za pravilno delovanje omrežja odločilnega pomena. Največja dopustna dolžina vodila pada z naraščanjem hitrosti prenosa. Pri hitrosti prenosa 1 Mb/s je le-ta največ 50 m, pri hitrosti 500 Kb/s ne sme biti več kot 100 m, in tako naprej.

Vsak okvir (ali paket) začanja z 11 bitno razpoznavno številko. CAN sicer ne uporablja naslovov vozlišč, vendar je z razpoznavno številko direktno določeno, komu je okvir namenjen. Za primer je na sliki 128 skiciran primer, ko vozlišče V1 (npr. merilnik hitrosti) odda okvir s številko 11100001111 (30F šestnajstičko), ki je namenjeno vozliščema V2 (npr. prikazovalniku hitrosti na armaturni plošči) in vozlišču V4 (npr. sistemu za regulacijo motorja). Ostala vozlišča okvir zavržejo. Ta odnos se določi v času konfiguriranja sistema.

6.3.1 Dostop do kanala v omrežju CAN

Sedaj pa bomo razložili princip, po katerem vozlišča v omrežju CAN dostopajo do prenosnega medija (sredstva). Ta je za CAN bistven in najbolj značilen. CAN realizira prioritetni CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) oziroma CSMA s preprečevanjem trčenj (nedestruktivni CSMA).



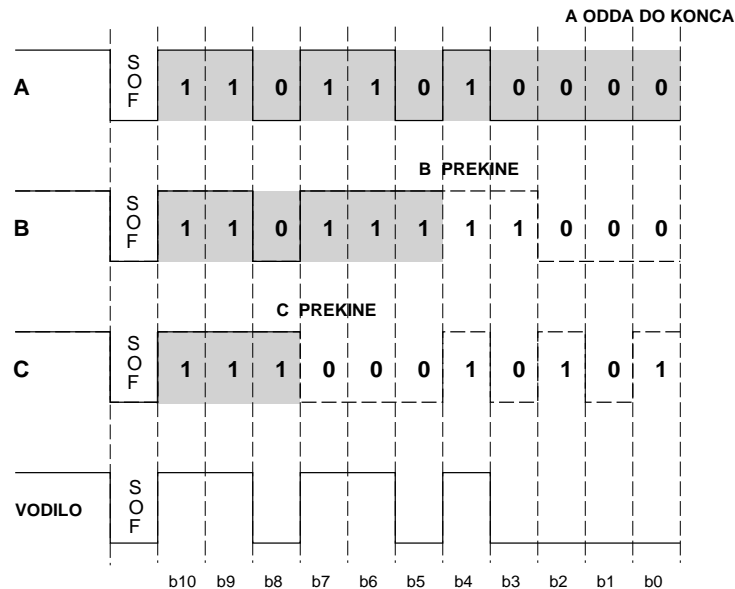
Slika 128: Topologija omrežja CAN in princip naslavljanja. Okvir je določen z razpoznavno številko. Vsa vozlišča, ki jih zanimajo podatki okvirja, lahko istočasno sprejmejo podatke.

Razpoznavna številka okvirja pomeni hkrati tudi njegovo prioriteto. Manjša vrednost (kodirana binarno, najpomembnejši bit se prvi odda) pomeni višjo prioriteto. Ker so kombinacije 111111xxxx, kjer je x 0 ali 1, zaradi kompatibilnosti prepovedane, je tako možnih 2032 prioritetenih nivojev. Vodilo (največkrat RS485) deluje kot krajevno porazdeljena logična vrata IN. V mirovnem stanju je na vodilu logična enica (visok nivo). V primeru, da začne sočasno z oddajo več vozlišč, bo na vodilu stanje ena samo v primeru, da vsa vozlišča oddajo enico. Če pa vsaj eno vozlišče odda ničlo, ničla prevlada in postavi vodilo v stanje nič. V tem smislu je logična ničla “prevladujoča” ali “dominantna”, enica pa “popustljiva” ali “recesivna”. Na tem je zasnovano preprečevanje trčenj. V primeru, da je vodilo prosto, lahko vsako vozlišče, ki ima željo po oddaji, začne takoj z oddajo. Če je takih vozlišč več, pride do trčenja. Vozlišča, ki začnejo sočasno z oddajo, so s tem sinhronizirana na začetek okvirja. Druga vozlišča, pri katerih pride do potrebe po oddaji kasneje, čutijo, da je vodilo zasedeno in počakajo z oddajo.

Poglejmo konkreten primer. Ne pozabimo, da postaje med oddajanjem tudi sprejemajo. Naj pride do sočasne oddaje naslednjih okvirjev:

	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Vozlišče A:	1	1	0	1	1	0	1	0	0	0	0
Vozlišče B:	1	1	0	1	1	1	1	1	0	0	0
Vozlišče C:	1	1	1	0	0	0	1	0	1	0	1

Po dogovoru ima najvišjo prioriteto okvir, ki ga odda vozlišče A, torej mora njen okvir “preživeti”. Slika 129 prikazuje razmere na vodilu. Vse tri postaje začnejo z oddajo dominantne ničle, ki označuje začetek okvirja. Ker so vse tri postaje začele sočasno z oddajo, so njihovi okvirji sinhronizirani. Nato vse tri postaje oddajo prvi prioritetni bit, ki je recesivna enica, zato je vodilo v stanju



Slika 129: Primer razrešitve problema sočasne oddaje. Vodilo dobi okvir z najvišjo prioriteto.

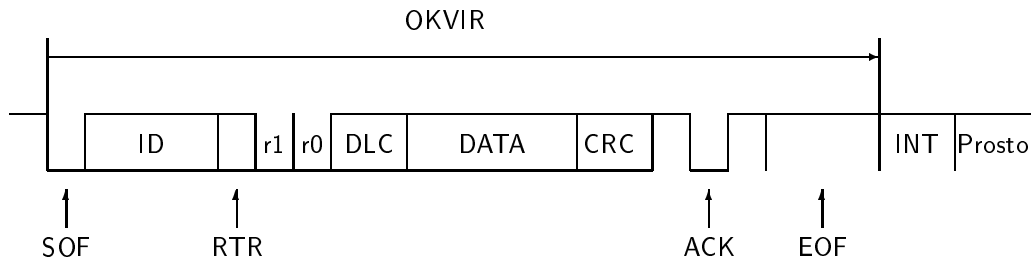
ena. Vsaka izmed postaj primerja sprejeto vrednost z oddano vrednostjo. Ker se pri vseh postajah sprejeta vrednost ujema z oddano, nadaljujejo z oddajo. Enako je po oddaji naslednjega bita. Pri oddaji tretjega bita prevlada ničla. Postaji *A* in *B* ponovno sprejmeta enako vrednost, kot sta oddali, zato nadaljujeta z oddajo. Postaja *C*, ki je enako kot *A* in *B* sprejela ničlo vendar oddala enico, pa takoj prekine z oddajo in se preklopi na sprejem. Od tu naprej tekmujeta za vodilo samo še postaji *A* in *B*, dokler ne oddata šestega bita. Postaja *A* tedaj odda dominantni bit, s čimer dokončno ostane edina oddajna postaja in odda vsebino celega okvirja brez prekinitve.

CAN uporablja NRZ obliko signala. V primeru daljšega zaporedja simbolov enake vrednosti se tak signal malo spreminja in možen je izpad sinhronizacije. Da do tega ne pride, se na vsakih 5 simbolov enake polaritete, na oddajni strani vrine polnilni simbol obratne polaritete, ki se na sprejemni strani avtomatsko izločijo.

6.3.2 Oblika okvirja v omrežju CAN

CAN pozna dve obliki okvirjev: standardni okvir (CAN verzija 2.0A) in “razširjeni” ali “podaljšani” okvir (CAN verzija 2.0B). Edina razlika je v dolžini identifikacijske številke. CAN 2.0B uporablja 29 bitno razpoznavno številko (11 bitna identifikacija in 18 bitni podaljšek), tako da se zagotovi kompatibilnost “za nazaj”. Večinoma za identifikacijo zadošča 11 bitov, zato se CAN 2.0A več uporablja. Obliko standardnega okvirja prikazuje slika 130.

Začetek okvirja (SOF - Start Of Frame) označuje dominantna ničla, t.j. prehod signala na vodilu iz mirovnega (recesivna enica) v aktivno stanje (domi-



Slika 130: Oblika okvirja protokola CAN.

nantna ničla). SOF služi za vzpostavitev začetne sinhronizacije, tako imenovane trde sinhronizacije, sprejemnih vozlišč. Ponovna sinhronizacija je možna med sprejemanjem okvirja v smislu skrajšanja ali podaljšanja trajana signala enega bita. Polje za arbitražo vodila obsega 12 bitov, od teh je prvih 11 bitov identifikacijskih oziroma prioritetnih (ID), zadnji pa je RTR bit (ang. Remote Transmission Request). S pomočjo tega bita lahko neko vozlišče zahteva od drugega (oddaljenega) vozlišča oddajo podatkovnega okvirja z enako razpoznavno številko kot je številka zahteve. Okvir z bitom RTR = 0 je običajen podatkovni okvir, medtem ko okvir z RTR = 1 ne vsebuje podatkov, temveč pomeni zahtevo za oddajo podatkovnega okvirja. Polje dolžine okvirja je v tem primeru nepomembno.

Sledi šestbitno kontrolno polje, pri čemer štirje biti (DLC) označujejo dolžino okvirja (število podatkov v okvirju), prva dva bita (r1, r0) pa sta dominantna in predvidena za kasnejše razširitve. DATA je podatkovno polje in vsebuje od 0 do 8 bajtov podatkov. Polje CRC (Cyclic Redundancy Check) vsebuje 15 bitov za preverjanje pravilnosti prenosa in en ločilni recesivni bit. Za preverjanje je izbran generatorjev polinom stopnje 15,

$$x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$$

CRC ostanek se računa od vključno SOF bita. Polje za potrditev sestavlja dva bita. Prvi je oddan kot recesiven, vendar ga kasneje vsa vozlišča, ki pravilno sprejmejo okvir, postavijo v dominantno stanje. Ta bit se imenuje potrdilna reža (ACK Slot). Drugi bit je ločilna recesivna enica. Na ta način se potrdilni bit nahaja med dvema recesivnima bitoma, kar poveča zanesljivost njegove detekcije. Konec okvirja (EOF - End Of Frame) sestavlja sedem recesivnih enic. Za koncem okvirja sledi časovni premor treh recesivnih enic (INT), potem pa začne nov okvir ali pa vodilo preide za poljubno dolgo v mirovno stanje.

6.3.3 Časovna analiza protokola CAN

V tem podglavju bomo ocenili odzivnost (reakcijske čase) za sporočila v omrežju CAN. Zanimalo nas bo, kako dostop do kanala po načelu CAN vpliva na odzivni čas, to je na čas, ki poteče od trenutka ko je dana zahteva za prenos nekega sporočila do tedaj, ko je sporočilo v celoti prenešeno. Protokol CAN predvideva fiksne prioritete sporočil. Zato ima sporočilo z višjo prioriteto vedno prednost pred sporočili z nižjo prioriteto in pride pri prenosu prej na vrsto. Pri prioritetenem načinu časovnega razvrščanja s fiksnimi prioritetaami se običajno privzame, da je zahtevano odzivnost možno zagotoviti samo sporočilom z najvišjo prioriteto. Ken Tindell [9] pa je pokazal, da se da na osnovi poznanih parametrov omrežja določiti zgornjo mejo odzivnega časa za vsako sporočilo, tudi za tisto z najnižjo prioriteto. V nadaljevanju bomo opisali Tindelov pristop.

Predpostavimo, da je v omrežju končno mnogo sporočil S_m in da je število vseh sporočil v omrežju znano vnaprej in enako M ,

$$\text{množica sporočil} = \{S_1, S_2, \dots, S_m, \dots, S_M\}.$$

Sporočila, ki se potegujejo za prenos, so monotono urejena po podajočih prioritetah. Sporočila so lako bodisi periodična bodisi sporadična (vezana na dogodke, ki se zgodijo nepredvidljivo), vendar je čas med zaporednima nastankoma istega sporočila navzdol omejen in poznan vnaprej. Torej privzamemo, da za vsako sporočilo poznamo ponovitveni cikel T_m . Poleg tega imamo za vsako sporočilo podan skrajni rok (ang. Deadline) D_m , v katerem mora biti dostavljeno (prenešeno do cilja), seveda pa je lahko prenešeno že prej. Naša naloga je, da določimo odzivni čas R_m za vsako posamezno sporočilo, in sicer v najbolj neugodnih okoliščinah. Če naj omrežje CAN zadosti zahtevam časovno kritičnih sistemov, mora zagotoviti skrajni rok dostave za vsako sporočilo, tudi za tisto z najnižjo prioriteto. Pogoji torej je:

$$R_m \leq D_m, \quad (m = 1, 2, \dots, M).$$

Če to v danih okoliščinah ni mogoče doseči, CAN omrežje ne izpolnjuje pogojev za delovanje v stvarnem času.

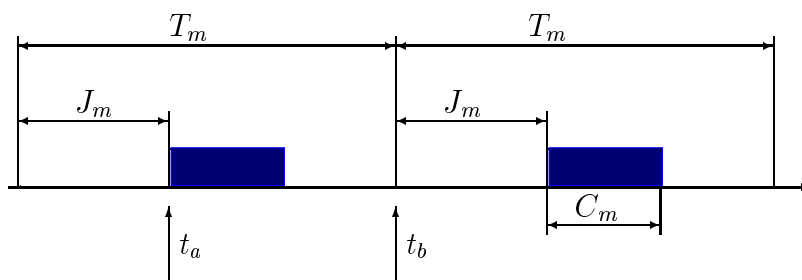
Slika 131 prikazuje model, s katerim si bomo pomagali pri analizi protokola CAN. Predpostavimo, da od trenutka, ko sporočilo S_j nastane, do trenutka, ko se pripravi (razvrsti) za prenos, mine največ čas J_m . S tem zajamemo variabilnost časovne razvrstitve za vsako posamezno sporočilo, ki se sicer s časom spreminja in je včasih lahko tudi nič. Imenujemo jo negotovost razvrstitve. Kot rečeno, za vsako sporočilo poznamo ponovitveni cikel T_m . Sporočilo S_m , ki čaka na oddajo, mora biti oddano pred prihodom naslednjega, ker se v nasprotnem primeru tekoče sporočilo v oddajnem medpomnilniku "prekrije" z novim sporočilom. Zaradi nedoločenosti razvrstitve je najkrajši interval med dvema zaporednima zahtevama za prenos enak $t_b - t_a$. V tem času mora biti sporočilo v najneugodnejših razmerah

oddano, sicer se izgubi. Iz tega sledi naslednji dodatni pogoj:

$$R_m \leq (T_m - J_m), \quad (m = 1, 2, \dots, M)$$

ali v splošnem:

$$R_m \leq \min\{(T_m - J_m), D_m\}, \quad (m = 1, 2, \dots, M).$$



Slika 131: Model za analizo protokola CAN.

Sedaj bomo določili odzivne čase R_m za najneugodnejše razmere. Tedaj je odzivni čas sporočila S_m je seštevek časa prenašanja sporočila C_m in časa čakanja na oddajo w_m . K temu moramo prišteti še čas potreben za razvrstitev J_m ,

$$R_m = J_m + w_m + C_m. \quad (35)$$

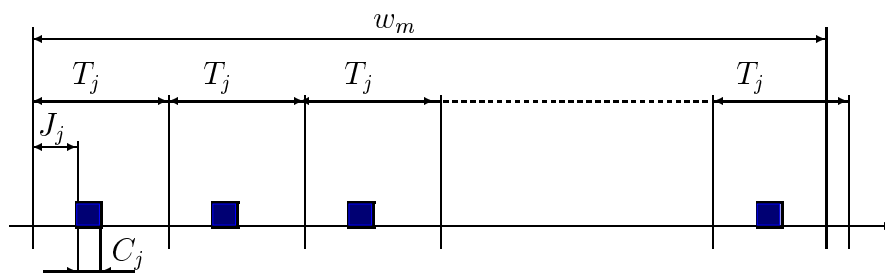
Za dano hitrost prenosa oziroma za dan čas trajanja enega bita τ je čas prenosa odvisen od dolžine – števila bitov v sporočilu. Dolžina sporočila CAN v bitih je $47 + 8 \times s_m$, pri čem 47 bitov prispevajo uvodni in zaključni biti in s_m je število podatkovnih bajtov znotraj okvirja, ki znaša največ osem. Ker CAN na vsakih 5 enakih simbolov (ničel ali enic) vrine simbol z drugačno vrednostjo (med enice ničlo, med ničle enico), bo v najslabšem primeru v sporočilo dodal še $\lfloor \frac{34+8 \times s_m}{5} \rfloor$ ($47-34 = 13$ je izvzetih iz polnjenja). Zato je

$$C_m = (\lfloor \frac{34 + 8 \times s_m}{5} \rfloor + 47 + 8 \times s_m) \times \tau. \quad (36)$$

Določiti moramo še čas čakanja na oddajo w_m . Ta je seštevek čakanja na konec prenosa tekočega sporočila, ki je v najslabšem primeru enak trajanju najdaljšega sporočila nižje prioritete B_m (čas blokade) in skupnega časa prenašanja sporočil z višjo prioriteto:

$$w_m = B_m + \sum_{\forall j \in hp(m)} n_j \times C_j, \quad (37)$$

pri čemer je $hp(m)$ množica sporočil z višjo prioriteto od sporočila S_m , C_j čas prenašanja sporočila S_j , z n_j pa smo označili število sporočil S_j , ki nastanejo med



Slika 132: Ponazoritev števila sporočil S_j v času w_m .

tem, ko sporočilo S_m čaka, da pride na vrsto. Razmere so ponazorjene na sliki 132.

Ker je ponovitveni cikel sporočila S_j enak T_j , se v času w_m zvrsti $\lceil \frac{w_m}{T_j} \rceil$ zahtev za prenos sporočila S_j . V resnici lahko zaradi razvrstitve negotovosti J_j v času w_m nastane celo $n_j = \lceil \frac{w_m + T_j}{T_j} \rceil$ zahtev za prenos. Vseh n_j sporočil mora biti prenešenih pred oddajo sporočila S_m . Če to upoštevamo v enačbi (37), dobimo:

$$w_m = B_m + \sum_{\forall j \in hp(m)} \lceil \frac{w_m + J_j}{T_j} \rceil \times C_j, \quad (38)$$

pri čemer je $hp(m)$ množica sporočil z višjo prioriteto od S_m . Neznanka w_m nastopa na levi in na desni strani enačbe in se je ne da izraziti eksplicitno. Zato enačbo rešimo numerično z iterativno metodo. Iteracijska enačba je:

$$w_m^{(k+1)} = B_m + \sum_{\forall j \in hp(m)} \lceil \frac{w_m^{(k)} + J_j}{T_j} \rceil \times C_j, \quad (k = 0, 1, 2, \dots), \quad (39)$$

pri čemer za začetni približek $w_m^{(0)}$ vzamemo na primer nič. Po enačbah (39) in (35) izračunamo čakalne in odzivne čase za vsa sporočila od najvišje do najnižje prioritete. Ker je čakalni čas w_m za sporočilo z nižjo prioriteto daljši od čakalnega časa sporočila z neposredno višjo prioriteto w_{m-1} , lahko v iteracijski enačbi za začetni približek vzamemo kar $w_m^{(0)} = w_{m-1}$, ($m = 1, 2, \dots, M - 1$).

6.4 Interbus

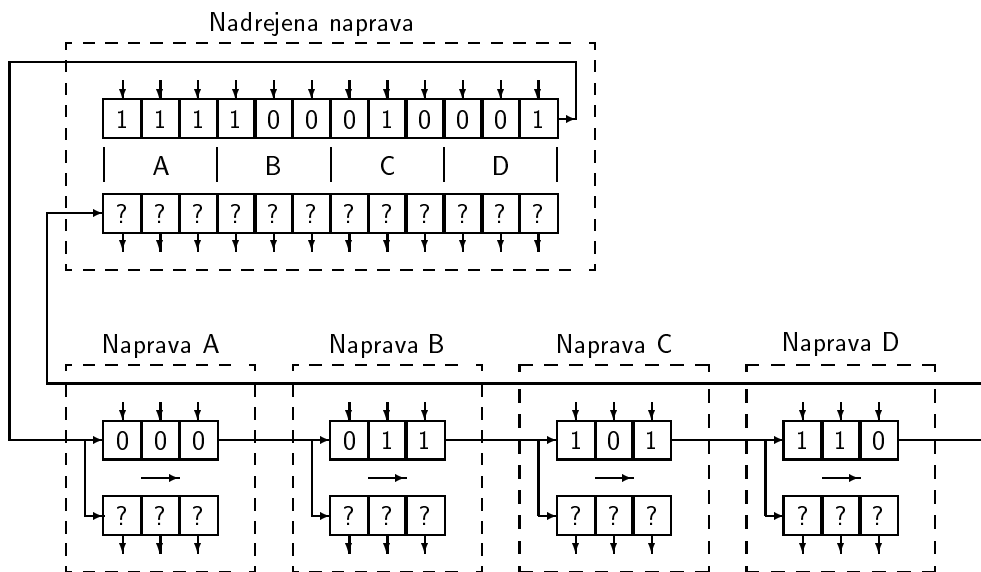
Interbus-S (IBS) je razvila firma Phoenix Contact za potrebe procesno merilne tehnike. Navzven je to omrežje sicer res videti kot vodilo, vendar je s topološkega vidika Interbus-S v resnici obroč. V obroču je vedno ena nadrejena naprava (ang. Master) in več podrejenih (ang. Slave). Nadrejena naprava (gospodar) skrbi za pravilno delovanje omrežja, t.j. poskrbi za vzpostavitev (inicializacijo)

in obratovanje omrežja. Podrejene naprave opravljajo vhodno izhodne funkcije: sprejemajo podatke iz procesa in krmilijo proces. Naprave so vezane zaporedno druga za drugo v verigo: nadrejena naprava je vezana na prvo podrejeno napravo, ta je vezana na naslednjo podrejeno napravo, i.t.d. Zadnja podrejena naprava je vezana ponovno na nadrejeno napravo. Delovanje omrežja si lahko ponazorimo z velikanskim pomikalnim registrom, ki je porazdeljen med naprave. V tem pomikalnem registru se podatki krožno pomikajo oziroma prenašajo odnaprave do naprave. Vsaka naprava ima svoj sprejemni in svoj oddajni pomikalni register: v oddajni register vpiše podatke, ki jih zajame iz procesa, po drugi strani pa poskrbi, da se podatki iz sprejemnega registra pravilno prenesejo na ustrezne krmilne izhode.

Kroženje (osveževanje) podatkov v obroču se ponavlja cel čas obratovanja omrežja. Časovnemu intervalu, v katerem so sprejeti in oddani podatki vseh naprav, pravimo cikel. Podatke, ki so oddani in sprejeti v enem ciklu, imenujemo okvir. Oddajno sprejemni (osvežitveni) cikel začne in konča nadrejena naprava. Nadrejena naprava javi podrejenim napravam začetek osveževalnega cikla, ki zato prepisejo zajete podatke iz procesa v svoje oddajne registre. Podobno nadrejena naprava prepise oddajne podatke iz svojega internega pomnilnika v oddajni pomikalni register in začne z oddajanjem. Podrejene naprave pošiljajo podatke iz svojega oddajnega pomikalnega registra naprej naslednji napravi. Tako pridejo ti podatki preko pomikalnih registrov drugih naprav končno v sprejemni register nadrejene naprave. Po drugi strani pa se podatki, ki jih pošilja nadrejena naprava pomikajo preko pomikalnih registrov podrejenih naprav do postaje, ki so ji namenjeni. Ko pridejo podatki po obroču do postaje, ki so ji namenjeni, jih ta prevzame (prepiše iz sprejemnega pomikalnega registra). Oddajanje in sprejemanje poteka sočasno. Ko je osveževalni cikel (in s tem okvir) končan, so v sprejemnih registrih vseh podrejenih naprav sveži podatki, v sprejemnem registru nadrejene naprave pa so podatki, ki so jih oddale podrejene naprave.

Princip delovanja omrežja bomo razložili na sliki 133. Slika prikazuje poenostavljeno vezavo naprav. Za vsako od štirih podrejenih naprav smo si zamislili 3 bite sprejemnih in 3 bite oddajnih podatkov. Toliko je tudi dolžina njihovih pomikalnih registrov. Dolžina pomikalnih registrov nadrejenih naprav je enaka vsoti dolžin registrov podrejenih naprav, v tem primeru torej 12. Toliko je tudi dolžina podatkovnega dela okvirja. Nadrejena naprava hrani oddajne in sprejemne podatke za vse podrejene naprave v svojem pomnilniku. Oddajne podatke vpiše v oddajni register. Ko je oddajno sprejemni cikel končan (oz. okvir oddan), se v sprejemnem pomikalnem registru nahajajo sprejeti podatki in nadrejena naprava jih prepise v svoj pomnilnik.

Slika 133 prikazuje vsebine registrov tik predno začne (naslednji) osveževalni cikel. V naslednji tabeli so zbrani podatki, ki jih želi nadrejena naprava oddati podrejenim napravam, medtem ko imajo podrejene naprave tudi pripravljene

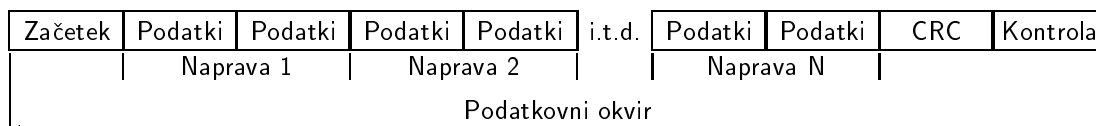


Slika 133: Delovanje omrežja Interbus.

podatke.

Postaja	Od nadrejene k podrejeni postaji	Od podrejene k nadrejeni postaji
A	111	000
B	100	011
C	010	101
D	001	110

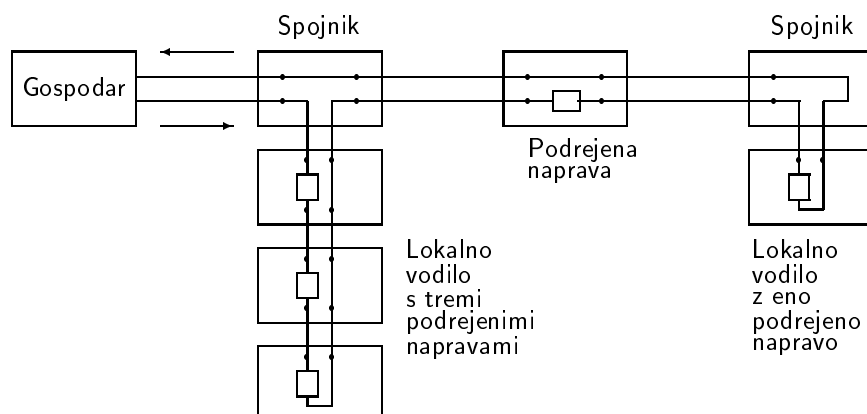
Vidimo, da je vsebina in pomen vsebine okvirja odvisna od vrstnega reda povezovanja postaj v obroč. Slika 134 prikazuje načelno obliko podatkovnega okvirja. Po najavi konca prejšnjega in začetka naslednjega okvirja so podatki za najbolj oddaljeno napravo, t.j. Napravo 1. Na tem mestu so podatki, ki jih ta naprava odda, pa tudi sprejme. Sledijo podatki drugih naprav v enakem zaporedju, kot so naprave vezane v obroč. Za podatki sta še CRC beseda za preverjanje pravilnosti beseda in kontrolna beseda (5 bitov). CRC računanje in preverjanje opravljajo vse naprave. Če ugotovijo napako, podatkov ne upoštevajo.



Slika 134: Izgled okvirja v omrežju Interbus-S

V začetku obratovanja nadrejena naprava ne pozna vrstnega reda naprav, zato začne identifikacijski cikel. S tem ciklom mora gospodar ugotoviti tipe posameznih naprav, njihov vrstni red kot tudi število aktivnih naprav v omrežju. V ta namen pošlje telegram, ki postavi vse podrejene naprave v ID stanje. Ta telegram sprejmejo hkrati vse naprave, kajti oddajni registri podrejenih naprav so v začetku premoščeni. Zato podrejene naprave pripravijo svojo razpoznavno številko (ID). Nato gospodar odda besedo za začetek okvirja (ang. Loop Back Word) in s tem začne ID cikel. V ID ciklu naprave vpisujejo (oddajajo) v okvir svoj ID, ki jih gospodar sprejema, dokler končno ne pride znak za začetek okvirja po obroču nazaj na okrog. S tem mu postane topologija obroča znana.

Za izvedbo omrežja Interbus-S sta predvidena dva tipa vodila: tako imenovano "daljinsko" vodilo in lokalno vodilo (slika135). Obe vodili prenašata signale z enakim pomenom, nivoji signalov pa so različni. Hitrost prenosa je na obeh vodilih 500 Kb/s. Daljinsko vodilo povezuje postaje na daljavo do 400 m, signali so v skladu s standardom RS485. Lokalno vodilo mora biti kratko (povezovanje modulov v enem ohišju). Služi istočasno za prenašanje podatkov in napajanje naprav. Lokalno vodilo se priključi na daljinsko vodilo preko prilagodilnih enot.



Slika 135: Izvedba omrežja Interbus-S. Na daljinsko vodilo vežemo lokalna vodila. Podrejene naprave vežemo direktno na daljinsko vodilo ali na lokalna vodila.

Literatura

- [1] James J. Pinto, "Fieldbus: A Neutral Instrumentation Vendor's Perspective", Action Instruments, Inc. San Diego, CA 92123, 1996.
- [2] J. Coutinho, S. Martin, G. Samata, S. Tapley, D. Wilkin, "Fieldbus tutorial", maj 1995.

- [3] Klaus Bender, "PROFIBUS Der Feldbus für die Automation", Carl Hanser Verlag München Wien, 1990.
- [4] Michael Volz, *PROFIBUS*, PROFIBUS Nutzerorganisation, Technische Druckschrift, 1994.
- [5] DIN19245 Teil 1, PROFIBUS, Process Field Bus, Übertragungstechnik, Buszugriffs- und Übertragungsprotokoll, Dienstschnittstelle zur Anwendung-Schicht, Management, Beuth Verlag GmbH, Berlin 1991.
- [6] K. Furman, *Analiza industrijskega procesnega vodila PROFIBUS*, Diplomsko delo, Fakulteta za elektrotehniko in računalništvo, Ljubljana 1992.
- [7] Anka Ščerk, *Industrijsko področno vodilo PROFIBUS*, Zaključna naloga, Fakulteta za elektrotehniko, Ljubljana 1996.
- [8] -, *CAN Specification, Version 2.0*, Robert Bosch GmbH, Stuttgart, September 1991.
- [9] K. Tindell, A. Burns, "Guaranteeing Message Latencies on Controller Area Network (CAN)", Proceedings 1st International CAN Conference, Mainz 1994.
- [10] K. Tindell, "Analysis of Hard Real-Time Communications", YCS-94-222, Department of Computer Science, University of York, England, 1994.